

HiSoft Pascal80

CP/M

£42.95
+VAT

HiSoft Pascal for Amstrad CP/M systems has been developed to provide an almost full implementation of Standard Pascal as detailed in the Pascal User Manual and Report by Kathleen Jensen and Niklaus Wirth (the originator of Pascal). We have been careful to adhere to this definition of Pascal while providing extra facilities to exploit your computer.

The compiler is written in Z80 and produces Z80 object code directly to an executable COM file (no P-codes or

linkage); this means that the compiler package, occupies very little space in memory (less than 19K) and produces object code that runs very quickly indeed. The minimum run-time overhead is about 2K.

Despite its compact size HiSoft Pascal is an extensive implementation of the Pascal language viz:

Predefined Identifiers

CONST

MAXINT = 32767;

TYPE

BOOLEAN=(FALSE, TRUE);

CHAR(The expanded ASCII character set of 256 elements)

INTEGER =-MAXINT..MAXINT

REAL(A subset of the real numbers with 7 digit accuracy);

TEXT=FILE OF CHAR;

VAR

INPUT, OUTPUT, TEXT;

PROCEDURE

READ;	READLN;	WRITE;	WRITELN;	PAGE;	HALT;	USER;
POKE;	INLINE;	NEW;	MARK;	RELEASE	OUT;	PRON;
PROFF;	RESET;	REWRITE;	GET;	PUT;	RANSEED;	CHAIN;
DISPOSE;						

FUNCTION

ABS;	SQR	SQRT;	ODD;	ENTIER;	ROUND;	TRUNC;
FRAC;	ORD;	CHR;	SUCC;	PRED;	EOLN;	EOF;
INCH;	INP;	PEEK;	ADDR;	CPM;	RANDOM;	SIN;
COS;	TAN;	ARCTAN;	EXP;	LN;	TRUNC;	SIZE;
RECAST;	MEMAVAIL;					

Reserved Words

AND	ARRAY	BEGIN	CASE	CONST	DIV	DO
DOWNT0	ELSE	END	FILE	FOR	FORWARD	FUNCTION
GOTO	IF	IN	LABEL	MOD	NIL	NOT
OF	OR	PACKED	PROCEDURE	PROGRAM	RECORD	REPEAT
SET	THEN	TO	TYPE	UNTIL	VAR	WHILE
WITH						

HiSoft Pascal80

The compiler supports all the data structures of Standard Pascal including Arrays, Sets, Records, Pointers, Text Files and user-enumerated types. Strings are supported as `ARRAY(1..N) OF CHAR`. Assignment of structured types, such as array-to-array assignment, is allowed and sets, strings and pointers may be compared. The `CASE` statement may optionally be terminated with `ELSE`. Procedures and functions may be fully recursive and procedures can take value or variable parameters. Records of any type may be created and accessed by field, optionally using the `WITH` construct for added convenience. Dynamic variables are created using the procedure `NEW` and accessed using pointers - this enables linked-list structures to be created. Further, the procedures `MARK`, `RELEASE` and `DISPOSE` allow control over the dynamic variable heap.

I/O is provided through the standard procedures `READ(LN)` and `WRITE(LN)`, `GET` and `PUT`. Standard CP/M text files, binary files or the console may be used as the input/output devices. Sequential files of any type are supported whilst random access can be achieved using Pascal source libraries supplied. Formatting of output is supported through the use of write parameters; reals may be output in scientific format (Mantissa, exponent) or fixed-point format.

Interfacing with Z80 machine code is made easy through `USER`, `PEEK`, `INLINE`, `SIZE` and `ADDR` together with register variables. `PEEK` and `POKE` may take any type as their arguments, except sets and files. `INLINE` allows machine code to be included in the program at runtime, and `ADDR` returns the address of a Pascal program variable. Interfacing to the Z80 ports is provided through the function `INP(P)` which reads port `P` and the procedure `OUT(P,C)` which outputs the value `C` to port `P`. The operating system may be accessed using `CPM(de,c)`.

Pascal80 programs can `CHAIN` another Pascal program, or any other CP/M program, preserving the global variable area. This, coupled with automatic saving of top-level procedures/functions during compilation, allows very large programs to be developed.

Many compiler options are available allowing, *inter alia*, output to be directed to a printer and runtime checks to be switched off for extra speed of execution. The 'F' option allows inclusion of source text from disc - this allows large programs to be compiled as well as providing the flexibility of storing program segments on disc.

Amstrad CP/M Plus (CPC6128, PCW8256/8512) versions of Pascal80 are supplied with full source libraries to allow access to the machine's GSX graphics interface. This allows lines to be drawn, points to be plotted and areas to be filled etc. Full documentation is supplied along with example programs and a new, enhanced screen driver for the PCW machines.

The compiler comes with an extensive ring-bound manual with example programs and some complete Pascal programs on the disc including an intelligent disassembler.

Also included in the package is our comprehensive screen editor, `ED80` (see `DEVPAC80` details for specification), which is fully integrated to allow a complete and easy development cycle. On a compilation or runtime error a helpful message is displayed, simply press a key and you will be returned into the editor with the cursor positioned near the offending symbol and with the message still displayed, ready to correct and re-compile.

The code produced by Pascal80 is compact (minimum size 2K), optimised and very, very fast.

Please study the above specification thoroughly; HiSoft Pascal, we believe, is the most powerful and the fastest Pascal compiler available for microcomputers at such a low price.